Vol.2011-HPC-129 No.15
2011/3/16

流体解析を対象としたAMG ライブラリの自動チューニング

藤 井 昭 宏[†1]　　中村 修[†2]　　小 柳 義 夫[†1]

SMAC 法による流体解析を対象に AMG ライブラリのオンライン自動チューニング手法を提案し，有効性を評価する．このような解析では，各時間ステップで圧力に関するポアソン方程式を解くことになる．この際に，AMG 法のパラメタの特性を考慮し探索範囲を狭め，パラメタを動的に選択する AMG ライブラリを作成した．ソルバのパラメタとして AMG のサイクル，各レベルの SOR 法の加速係数，AMG と組み合わせる複数の解法（GMRES 法，BICGSTAB 法，CG，組み合わせない）を考慮している．組み合わせると 147 個のソルバの中からを状況に応じて選択を行っていることとなる．本手法では途中解法が発散しても，動的に別の解法が選択されるため，解法を固定化している場合と比較し，解析をより安定に進めることができる．またソルバ全体の時間としても，自動チューニングを行わない場合と比較して最大 9 %程度性能が改善した．

Automatic tuning of AMG solver for fluid analysis

Akihiro Fujii,[†1] Osamu Nakamura[†2]
and Yoshio Oyanagi[†1]

This paper proposes the online automatic tuning method of AMG library for fluid analysis based on SMAC method. In this analysis, Pressure Poisson equation needs to be solved at each time step. We implemented the AMG library which determines a solver parameter adequately and effectively by assuming some behaviors of AMG parameters. The parameter is selected from combination of AMG cycles, acceleration coefficients for SOR at each level of AMG, and solvers preconditioned by AMG. Thus, this library selects a parameter setting among 147 solver settings. In our numerical tests, our memthod improved the performance of representative solvers up to 9 percent.

†1 工学院大学 Kogakuin University
†2 住友金属工業 Sumitomo Metal Industries

1. はじめに

近年，複数のライブラリ[1),2)]において自動チューニング機能が研究され，実装されているが，線形解法ライブラリにおいて問題や収束条件に応じて最適なパラメタ・解法を効率よく選択する技術はまだ確立していない．そのため，線形問題を繰り返し解く数値シミュレーションでは，多くの場合，適切であろう解法を選び，同じパラメタですべての問題を解くことが多い．本稿の目的は，このような問題の 1 例として流体解析を取り上げ，解析を進めながら解法やパラメタを動的に探索，設定することにより安定性を増しながら，全体のシミュレーション時間の削減することである．

本稿では SMAC 法による流体解析を研究の対象とする．この解析では圧力のポアソン方程式を各時間ステップで解くことになるが，各時間ステップで問題行列は変わらないことが多く，その場合，別々の時間ステップでの収束時間を比較することにより，適用された線形解法のパラメタの有効性の比較を行うことができる．そこでこのような解析を対象に各時間ステップで，適用した線形解法の種類，そのパラメタ，収束の条件，収束に要する時間を計測しながら，パラメタを評価していくことにより，パラメタを動的に選択する線形解法ライブラリを構築した．考慮するおもな解法は，AMG 前処理付 GMRES 法，CG 法，BICGSTAB 法と AMG 法単体であり，AMG 法のパラメタはマルチレベルの遷移のさせ方と各レベルの反復解法の加速係数である．組み合わせとして 147 個のソルバから状況に応じて適したものを選択していることとなる．

数値実験により，初めに最適な解法を選択せずとも，自動的に最適な解法が選択され，代表的な解法を固定的に実行した場合に比べて，最大９%速度が向上する場合があることを確認した．

以下 2 章で，SMAC 法の流体解析を紹介し，収束条件などを説明する．3 章でパラメタの評価・選択手法を提案し，4 章では 4 つの問題に対して数値実験を行う．5 章で関連研究に言及し，6 章でまとめる．

2. SMAC 法による流体解析

ここでは SMAC 法の簡単な紹介と，線形解法への収束条件の与え方，問題行列の逆行列がなく収束が不安定であった場合の対応法を説明する．

SMAC 法による流体解析は非圧縮性流体に対する手法であり，広く利用されている．次の時間ステップの流速 u^{n+1} を n ステップ目から直接計算せず，仮の流速 u^* を式 (1) によ

り u^n, p^n から求め，これを補正して u^{n+1} を計算する手法である．n+1 ステップでも非圧縮の条件 $\nabla \cdot u^{n+1} = 0$ が成立するようにするために，圧力の差分 $\varphi = p^{n+1} - p^n$ を式 (2) により求める．φ が求まると式 (3) により流速 u^{n+1} を計算する．この処理を繰り返し，時間ステップを進めていく．

$$\frac{u^* - u^n}{\Delta t} + (u^n \cdot \nabla)u^n = -\nabla p^n + \frac{1}{Re}\Delta u^n \tag{1}$$

$$\Delta\varphi = \frac{\nabla \cdot u^*}{\Delta t} \tag{2}$$

$$u^{n+1} = u^* - \Delta t \nabla \varphi \tag{3}$$

式 (2) に対して線形解法を適用することになる．この式の右辺は，有限体積法では各格子体積で積分した後，格子の体積で割ることにより計算できる．ただ、格子体積が不均一な場合に，単位体積当たりの式にしてしまうと左辺の問題行列が非対称になってしまう．本稿では問題行列の対称性を保つため，格子体積では割らないことにし，収束条件を残差ベクトルの閾値ベクトルとして格子体積に比例して条件を設定することとした．

また式 (2) の方程式を離散化した問題行列は逆行列を持たないことが多く，右辺ベクトルや反復解法によっては収束状況が不安定になる．本稿で対象とするある問題（ccFix）では，CG 法で発散してしまったため，以下のように安定化させた．この問題では定数ベクトルのみが 0 固有値ベクトルになることを利用し，問題行列 A の代わりに $A - ee^T$ を用い，解からは定数ベクトル成分を削除した．ここで e は 2 ノルムが 1 の定数ベクトルを表す．またこの問題では BICGSTAB 法や GMRES 法では各反復で解から定数ベクトル成分を削除することのみ行うこととした．

3. 自動チューニング手法

3.1 概　　要

本稿では SMAC 法による流体解析を対象にしている．各時間ステップで解く問題はポアソン方程式であり，時間ステップにより変化しない場合を想定している．解法や AMG パラメタの設定は問題行列に依存するが，同じ問題を繰り返し解くという解析の性質から時間ステップを進めながら，最適化を進めていく手法を検討した．図 1 に本稿の自動チューニングの疑似コードを示す．

問題行列と収束条件の残差閾値ベクトルは変化しないため，解析を進める前に problem_store 関数と set_terminal_cond_vec 関数でライブラリ側にそれぞれ登録する．その後解析を進め定期的に性能を計測してパラメタを変更していく．

keisoku_fl が真のときは，収束条件までの相対残差を変数 rres に計算し，record_time 関数により収束時間の記録を行う．choose_solver 関数により，解法やパラメタを設定する．ここでパラメタ最適化中は確率的に多様なソルバが呼び出されるが，最適化後は一定の相対残差に対する収束時間が最短のものに，設定される．その後，solve 関数を呼び出すことにより，設定されたソルバで解を求める．初期の相対残差を求め，収束時間を記録する計算もすべての時間ステップで行うと負荷が大きくなるため，keisoku_fl を check_conv 関数で定期的に解除することで，負荷の低減を図っている．

また反復回数や収束状況は計算負荷なく計測できるため，check_conv 関数では，反復回数が急増したり，収束しなくなった場合は keisoku_fl をセットし，即座に，ソルバの最適化を開始することで，ソルバの安定性を高めている．

3.2 チューニングパラメタ

線形問題の解法として，AMG 前処理付解法を想定している．パラメタ選択の範囲は以下のもので，147 通りのソルバの中から選択していることとなる．

- AMG 前処理を付ける解法
  - GMRES(2)，GMRES(4), GMRES(6), GMRES(10),
  - BICGSTAB
  - CG
  - なし：AMG 法単体で適用
- AMG 法内部のパラメタ
  - サイクル：V サイクル，W サイクル，F サイクル
  - 各レベルの緩和法 SOR 法の加速係数：0.8，1.0，1.2，1.4，1.6，1.8，2.0

3.3 事 前 知 識

等方性の 3 次元のポアソン問題にて，AMG のパラメタを変化させた場合の AMG 前処理付解法の性能推移を図 2 に示す．図 2 では，AMG のデフォルトの設定（図 2 の 2 番のパラメタ）での解法の性能順位が，AMG を最適化した後の解法の順位と変わっていないことがわかる．経験的にも，デフォルトの AMG の設定でその解法の性能は評価できることが多いため，本稿でも最適化後の解法の性能順位はデフォルトの設定での性能の順位と変わらないと仮定した．

また AMG の SOR の加減速係数については，加速しすぎると発散することがしられているが，加減速の係数とサイクルによる性能の影響は分離できるかどうかは自明ではない．本

Vol.2011-HPC-129 No.15
2011/3/16

```
keisoku_fl = .true.                    !C はじめに keisoku_fl をセットする
call problem_store(A)                  !C 問題行列 A の線形解法ライブラリへ登録
call set_terminal_cond_vec(vec)        !C 残差の閾値ベクトルの設定

do t = 1, timesteps
   ..... !C 流体解析コード

  !C -- 線形解法部: Ax=b を解く
  if(keisoku_fl) then
    call relative_res_to_terminal_cond(rres) !C 収束条件までの相対残差の計算
  end if

  call choose_solver(keisoku_fl)       !C 平均収束時間の速いソルバ選択
  if(keisoku_fl) call system_clock(t1) !C 時間計測開始
  call solve(x,b,iter)                 !C 解法の呼び出し
  if(keisoku_fl) then
    call system_clock(t2)              !C 時間計測終了
    call record_time(t2-t1,rres)       !C 収束までにかかった時間を記録
  else
    call check_conv(iter, keisoku_fl)  !C 収束状況のチェックと keisoku_fl の管理
  end if

  ..... !C 流体解析コード
end do
```

図 1 自動チューニングライブラリ呼び出しの疑似コード

稿では，探索範囲の削減のため，加減速係数とサイクルの種類は分離して最適化できると仮定した．

本稿では利用していないが，解析の残りの時間ステップ数などの情報も活用することは可能である．

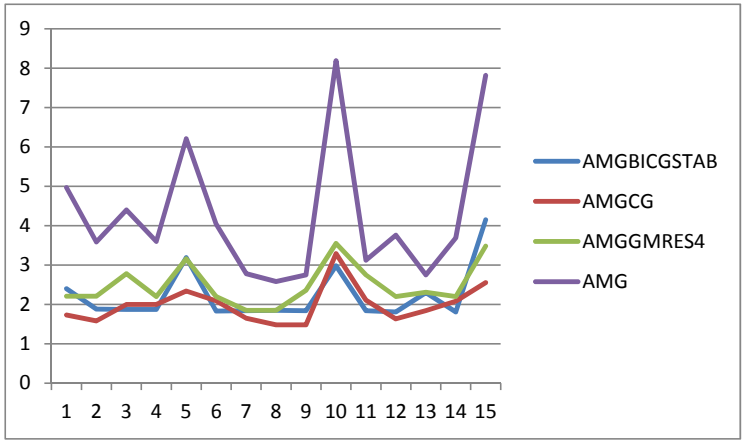

図 2 AMG 法のパラメタと AMG 前処理付解法の性能変化
縦軸：収束までの時間 [秒]，横軸：パラメタの種類，1-5, 6-10, 11-15 が V,F,W サイクル．各サイクルについて加速係数を 0.8, 1.0, 1.2, 1.4, 1.8 の順で計測．

3.4 目的関数と制約条件

自動チューニングの目的関数は全実施の合計コストである．また線形解法ライブラリの制約条件としては，以下の 3 つがあげられる．

- 収束条件が残差ベクトルの閾値ベクトルの形で与えられる
- 収束判定が間違っていても，解析コードの方で時間ステップを進めずに再度ソルバを呼び出す機能があるため，収束が難しい場合は早めにそのパラメタの実行を取りやめることも可能
- ソルバは 3.2 節にて記述した解法，パラメタの中から選択する

3.5 パラメタ選択手法

3.2 節から 3.4 節で仮定したことや条件から，以下の手順でのパラメタを選択する手法を実装した．

(1) デフォルトのパラメータでの AMG 前処理付解法の評価，解法の決定
(2) AMG の各レベルの反復解法の加速係数を変化させ，最適な加速係数を選択
(3) AMG のサイクルを変化させ最適なサイクルを選択
(4) 時間計測をせずに，(1)-(3) にて設定されたパラメタ，解法にて問題を解く

(5) (4) を一定回数実行後，ほかのパラメタ，解法を確率的に試行し，パラメタの最適性を確認する．確認できれば (4) へ．もし他のより高速な解法が計測されたら，(1) へ戻り，解法とパラメタを選択しなおす．

上記手順の (1)-(3) と (5) はパラメタ最適化中であり，パラメタの評価のためソルバが呼び出されたときの収束条件までの相対残差と収束に要した時間を記録している．この計測にも，計算負荷がかかるため，一度解法を最適化した後は，一定の時間ステップの間，手順 (4) のように時間計測を行わないこととしている．手順 (5) では，解法やパラメタの選択が適正かどうか確認するため，記録されている性能に比例して確率的にパラメタを変更して実行し，これまでの解法とは別の解法がより高速になれば，再度手順 (1) から最適化しなおす設定とした．

手順 (4) では，反復回数や収束状況は計測しており，反復回数が急増した場合や，収束しなくなった場合は即座に手順 (1) に戻り，パラメタの再設定を開始することで，解法の安定性を高めている．

このパラメタ選択手法を行う上での主なコストは以下の 2 点となる．

- パラメタ最適化中には時間ステップごとに初期残差を計算するため，行列ベクトル積が 1 回増える
- 性能による重みづけをしながら確率的に複数の解法を試行するため，性能の悪いものを選択する可能性がある

前者はパラメタ最適化を行う間隔を，後者は (5) における試行の回数を変更することで調整できる．当然ここで，性能改善幅とのトレードオフの関係となる．

4. 数値実験と考察

AMG法のデフォルト設定での解法と自動チューニング付の解法との性能比較について結果を紹介し考察する．

4.1 問題と計測条件

表 1 のような問題に対し，ソルバの評価を行った．安定化が○になっているものは，収束状況が不安定であったため，2 章にも記述した 0 固有値ベクトルが定数ベクトルであることを利用して安定化させたものである．時間ステップ数はどの問題でも 1000 ステップまで計算することとした．

また時間計測では，解析全体で線形解法のライブラリが要した合計時間で比較する．3 回試行し，その平均時間で比較を行った．また収束条件は残差ベクトルの各要素の大きさの閾値として与えられる．GMRES 法では残差ベクトルが毎回の反復で取り出せるわけではないため，GMRES 法の場合のみ相対残差により仮の収束条件を与え，その条件が満たされたときに，残差ベクトルを計算し，再度収束条件が成立するか判定している．

表 1 問題

問題名	cav	cavL	cc	ccFix
問題行列のサイズ	5000	100000	57420	57420
安定化	-	-	-	○
備考	キャビティフロー	キャビティフロー	6 面体非正方格子問題 1	6 面体非正方格子問題 2

3.5 節で説明したように，自動チューニング版では選択された解法が一定の時間ステップはその設定を利用し続ける．本稿ではその回数を 100 回とした．またパラメタの最適性の確認のために確率的に多様な解法が呼び出されるが，このためには 10 回試行することとした．パラメタの最適化中では，収束に要する時間を計測するが，各パラメタに対して残差ノルムが 1/10 になるまでの時間として補正して記録する．この値の過去 5 回分の平均をとってそのパラメタの収束時間としている．最終的にこの時間が最短のものが最も適したパラメタとして設定される．

4.2 実験結果と考察

図 3 にデフォルトの AMGCG 法のソルバ合計時間を 1 としたときの，自動チューニング版の解法（AT），AMG 法，AMGBICGSTAB 法，AMGGMRES(4) 法の相対時間を示す．また，表 2 には各解法を選択したときの線形ソルバ部分全体の時間と AMGCG 法の時間ステップあたりの平均反復回数を記述した．

図 3 から AT はより実効速度の高い解法を選ぶ手法のため，どの問題でも比較的良い性能を示している．また表 2 から問題 cc ではどの解法よりも 9 %程度高速になっている．cav では AT の性能があまり出ていないが，これは 1 タイムステップあたり，AMGCG 法が平均で 2.3 回程度で収束する問題であり AT のためのコストに見合う性能改善の余地が小さすぎることが原因と考えられる．図 4 に問題 cc のときに性能の高いＡＭＧＣＧ法とＡＴについて各時間ステップで線形ソルバ部分はどの程度時間がかかっていたか，またＡＴはどの解法を選択していたかを示す．ＡＴのグラフにおいて，解法の設定の書いていない区間については最適化中であり，確率的にさまざまな解法を呼んでいる．解法の設定が書かれている区間では，固定したパラメタでソルバが呼ばれており，解法，サイクルの種類，加速係数を示した．ＡＴの最適化中はＡＭＧＣＧ法と比較して時間ステップあたりの時間が多くかかって

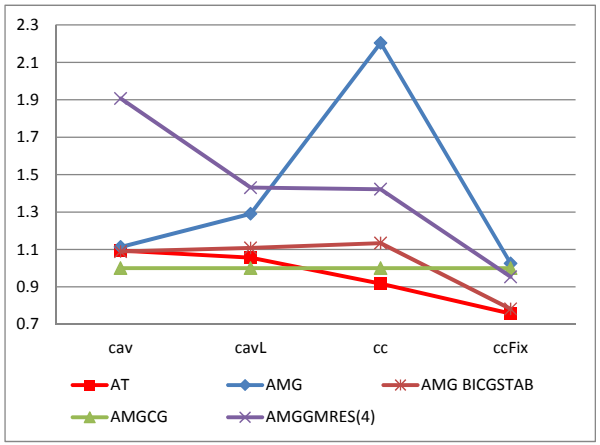

図 3 線形ソルバの性能比較
AMGCG 法の時間を基準としたときの各解法の相対時間

いること，またこの問題では，加速係数の最適化が有効に働いていることが分かる．

図 5 には各時間ステップでどの程度の時間がかかったか，を AT 以外のソルバについて示している．同じ問題，同じ時間ステップでも，同じ問題とはならないが，性質の近い問題になることが想定され，収束時間についての傾向は読み取ることができる．ここで cavL のグラフを見てみると，はじめ 40 から 80 ステップまでは AMGBICGSTAB 法が最も早いが，それ以降は AMGCG 法，AMGBICGSTAB 法が同等の速さになり，500 ステップ以降は AMGCG 法が最も高速になっていることがわかる．また cc では全体を通して AMGCG 法が，ccFix では AMGBICGSTAB 法が最も高速になっている．このように，問題を通じて，最適な解法が異なることもあり，時間ステップの途中で変わることもある．また ccFix 問題に対する AMGCG 法のように途中発散した場合も，他の適切な解法に動的に切り替えて対応ができる，という観点からも AT の有効性は高いと考えられる．

5. 関 連 研 究

直野ら[3] や櫻井ら[4] は，使用メモリ量や精度を数値計算ポリシーとして指定することでユーザからの利用しやすい，高性能なライブラリ方式について提案している．これは，ユーザが指定した条件の中で，実行時間を最小化することを目指しており，本研究の目指してい

表 2 問題ごとの各解法部全体の時間 [秒] と AMGCG 法の平均反復回数

問題	cav	cavL	cc	ccFix
AMGCG の平均反復回数	2.3	5.4	8.85	9.7
AT	6.63	262.5	323.0	278.3
AMG	6.75	320.7	776.6	377.1
AMGCG	6.06	248.5	352.3	367.6
AMGBICGSTAB	6.60	275.4	399.4	287.4
AMGGMRES(4)	11.57	355.5	501.2	350.6

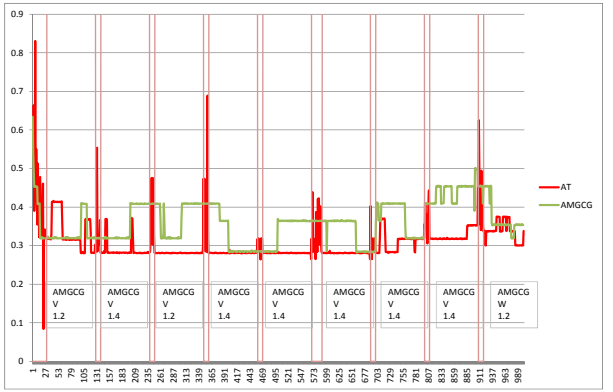

図 4 問題 cc におけるパラメタの切り替えと試行の様子
横軸：時間ステップ番号，縦軸：時間ステップあたりの線形ソルバの時間 [秒]

るところと似ている．また片桐ら[1] は実行環境に応じて適切な実装を選択する線形解法のライブラリの開発を進めている．しかし本研究は，アプリケーションを特定し，動的なパラメタ最適化について分析しており，その点では異なる．

このパラメタの動的な最適化という観点からは，須田[5] により自動チューニングの数理的性質がまとめられている．本研究でも「無限希釈」という考え方に基づいて 3.5 章にてパラメタ設定手法を設計した．

Chan ら[6] はポアソン方程式に対するマルチグリッド法で，精度に応じてサイクルの形を動的計画法によって最適化する手法を提案している．この研究は静的なパラメタ最適化手法であり，オフライン自動チューニングにあたる．本研究とは考察の対象が異なっている．また，Chan らは，要求精度によって最適なマルチグリッドサイクルの形が変化することを示していた．このことから本研究で扱う，AMG 前処理付解法でも同様に要求精度ごとにパ

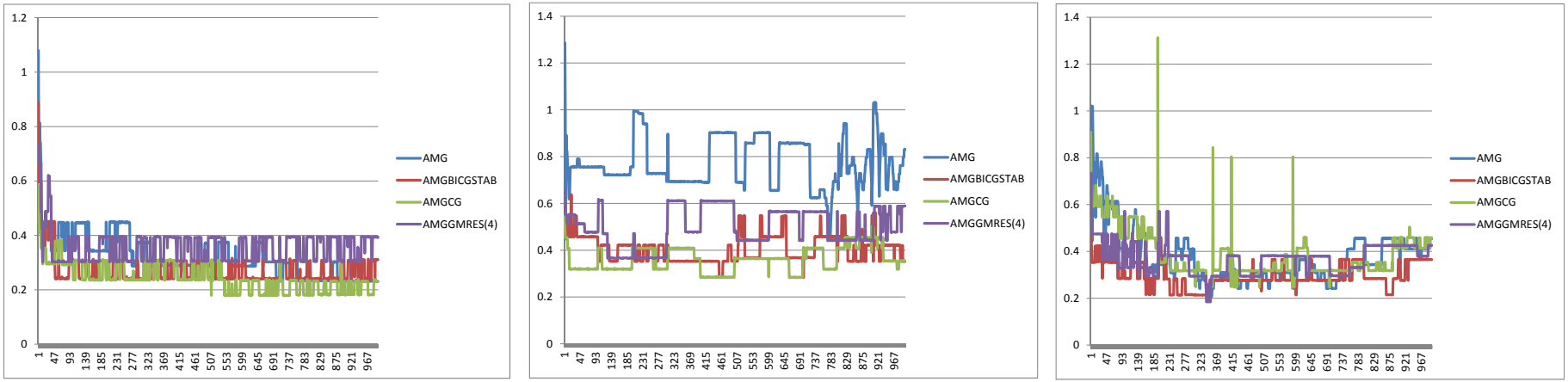

図 5 各時間ステップでのソルバーの時間, 左 cavL, 中央 cc, 右 ccFix
横軸：時間ステップ番号，縦軸：時間ステップあたりの線形ソルバの時間 [秒]

ラメタ最適化をした方が有利になることが予想されるが，有効性について実験からは示せなかった．この点については，計測する問題を増やして今後考察を深めたい．

6. ま と め

本研究では SMAC 法による流体解析を対象に，AMG 前処理付線形解法のパラメタの動的設定手法を提案し，有効性の評価を行った．その結果，提案手法は 1000 時間ステップの問題に対して最大 9 %程度性能が向上する場合があることが分かった．途中で解法が発散したり不安定になった場合にも他の解法にすぐに処理が移るため安定性も増していると考えられる．また，数値実験の中では，対象とする流体解析において代表的な線形解法の挙動を分析し，問題よって最適な解法が変化すること，解析の時間ステップによって適した解法が変化することがある場合を示した．

本稿でのパラメタ最適化手法はＡＭＧ法の一部のパラメタのみしかパラメタ選択の対象としていれなかったため，今後このパラメタを増やすと同時に，様々な環境での最適化も視野に入れて研究を進めていきたい．

参 考 文 献

1) Katagiri, T., Kise, K., Honda, H., and Yuba, T : ABCLib_DRSSED: A Parallel Eigensolver with an Auto-tuning Facility, Parallel Computing, 32, 3, pp. 231-250(2006).
2) Whaley, R.C., Petitet, A., and Dongarra,J.J.: Automated Empirical Optimizations of Software and the ATLAS Project, Parallel Computing, 27, pp.3–35(2001)
3) 櫻井隆雄，直野健片，桐孝洋，中島研吾，黒田久泰，猪貝光祥：行列計算ライブラリ向け数値計算ポリシーインターフェースの提案，情報処理学会 研究報告 (HPC), Vol.2010-HPC-126, No.42, pp.1–9 (2010).
4) 直野健，猪貝光祥，木立啓之：数値計算ポリシーを入力とするベクトル群の直交化ライブラリ，情報処理学会論文誌 (ACS)， Vol.46, No.SIG7, pp.35–43(2005)
5) 須田礼仁：自動チューニングのための数理基盤技術（数値計算のための自動チューニング），応用数理，日本応用数理学会， Vol.20, No.3, pp.191–200 (2010).
6) Chan, C., Ansel, J., Wong, Y.L, Amarasinghe, S., and Edelman, A. : Autotuning multigrid with PetaBricks, Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, SC '09, Portland, Oregon. No.5, pp. 1–12 (2009).